iBGP2: un mécanisme de redistribution iBGP menant à un routage optimal

Marc-Olivier Buob [†1] et Anthony Lambert[2] et Steve Uhlig[3]

[1]*Nokia Bell Labs, France*
[2]*Orange Labs, France*
[3]*Queen Mary University of London, United Kingdom*

L'Internet est consistué de plus de 50 000 ASes (Autonomous Systems) échangeant des informations de routage grâce au protocole BGP (Border Gateway Protocol). Au sein d'un AS, l'information est redistribuée via des sessions iBGP (internal BGP), permettant à chaque routeur d'associer toute destination extérieure à l'AS à un point de sortie. Les approches existantes (le full-mesh iBGP, la réflexion de route, et les confédérations BGP) ne permettent pas de garantir un routage optimal et de passer à l'échelle simultanément. Cet article propose un nouveau mécanisme de redistribution iBGP, appelé iBGP2 qui concilie ces deux aspects en permettant à chaque routeur de déterminer l'information de routage pertinente à transmettre à chacun de ses voisins. Notre contribution est triple. Tout d'abord, nous démontrons que notre mécanisme, iBGP2, conduit toujours à un routage stable, déterministe, correct et optimal. Ensuite, nous fournissons une implémentation open-source basée sur Quagga d'iBGP2. Enfin, nous montrons qu'iBGP2 est une solution crédible au travers de simulations effectuées sous ns-3. Une version étendue de cet article a été publiée à Infocom'2016.

Mots-clefs : BGP, protocoles de routage, algorithme de Dijkstra

1 Introduction

L'Internet est composé de plus de 50 000 systèmes autonomes (AS). Les routeurs situés en bordure d'AS s'échangent des informations de routage au travers de sessions eBGP (external Border Gateway Protocol). À l'intérieur de l'AS, cette information de routage est redistribuée au travers de sessions iBGP (internal BGP). BGP permet à chaque routeur d'élire une route pour chaque destination extérieure à son AS en lui associant un point de sortie. Ce point de sortie est en général routé grâce à un IGP (Interior Gateway Protocol). L'IGP permet de router les destinations internes à l'AS. Dans un IGP à état de liens, tel qu'OSPF ou IS-IS, chaque routeur maintient à tout instant un graphe orienté pondéré représentant le réseau, et déduit grâce à l'algorithme de Dijkstra ses plus courts chemins vers les destinations internes à l'AS.

Ce papier se focalise plus particulièrement sur la redistribution iBGP. Un routeur ne retransmet pas une route apprise d'un voisin iBGP à ses voisins iBGP. C'est pourquoi dans les petits AS, les opérateurs déploient généralement un *full-mesh iBGP*. Il consiste à configurer une session iBGP entre chaque couple de routeurs BGP de l'AS. Cette architecture passe toutefois difficilement à l'échelle, car chaque session iBGP est coûteuse en mémoire et en CPU.

Dans les AS de grande taille, les opérateurs ont principalement recours aux *réflecteurs de route* (RR). Un RR est un routeur BGP, qui, exceptionnellement, peut redistribuer certaines routes apprises par iBGP à des voisins iBGP. Certains de ses voisins iBGP sont déclarés "RR-clients". Un RR peut redistribuer à un RR-client (resp. voisin iBGP) une route qu'il a sélectionnée et apprise d'un de ses voisins iBGP (resp. RR-client). Les opérateurs définissent habituellement une hiérarchie de RR capable de redistribuer l'information BGP dans tout l'AS. Une alternative aux RR consiste à subdiviser un AS en sous-AS, appelés confédérations BGP. Elles sont peu utilisées car elles engendrent un routage interne moins réactif.

[†]Une partie de ces travaux a été effectuée au sein du LINCS (Laboratory of Information, Networking and Communication Sciences).

Qu'il s'agisse des RR ou des confédérations, le passage à l'échelle est rendu possible en filtrant une partie de l'information de routage à laquelle chaque routeur aurait été confronté dans un full-mesh. Malheureusement, ce filtrage peut parfois empêcher un routeur d'apprendre son meilleur point de sortie, conduisant alors à un routage sous-optimal. La littérature a montré que les RR pouvaient aussi conduire à un routage instable, non déterministe, ou même à des boucles de routage [BMU07, GW02a, GW02b]. [BMU07] montre toutefois que si chaque routeur apprend et sélectionne son meilleur point de sortie possible, le routage est toujours stable, sans boucle, optimal et déterministe. Une telle topologie iBGP est dite *fm-optimale*, car le routage est alors conforme à celui qu'on aurait obtenu avec un full-mesh.

Concevoir une topologie de RR minimale (en nombre de sessions iBGP) et fm-optimale est généralement complexe [BUM08]. De plus, la topologie calculée est remise en cause en cas de changement de métriques IGP. Les solutions existantes atteignant leurs limites, on se propose ici de revisiter la manière dont l'information est redistribuée via iBGP, afin de garantir un routage fm-optimal quelle que soit la topologie IGP, et ce y compris en cas de changement de topologie IGP.

2 Fm-optimalité

Le graphe IGP et iBGP sont des graphes orientés, tels que si un arc (u,v) existe, alors l'arc (v,u) existe aussi. Le graphe IGP est pondéré. Les métriques IGP de (u,v) et (v,u) sont indépendantes. On note $|u,v|$ la longueur du (d'un) plus court chemin IGP allant de u à v. Deux sommets peuvent être voisins dans le graphe iBGP sans l'être pour autant dans le graphe IGP (et réciproquement).

Dans tout l'article on considère implicitement une destination arbitraire D, extérieure à l'AS, pour laquelle plusieurs points de sortie sont envisageables. Les routes BGP qui en découlent sont dites *concurrentes*. De plus, on suppose que les routeurs élisent le point de sortie le plus proche (au sens IGP) parmi ceux qu'ils apprennent. Les routes BGP concernées sont dites *quasi-équivalentes*. Cette situation est très fréquente dans les réseaux formant le cœur de l'Internet. C'est aussi celle où surviennent les problèmes évoqués dans [BMU07, GW02a, GW02b] ‡. C'est pourquoi notre article se place dans ce cas. On note $\mathcal{S}$ l'ensemble des points de sortie, et $\mathcal{T}$ l'ensemble des routeurs de l'AS.

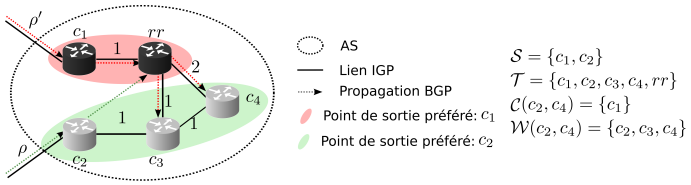

FIGURE 1: c_3 et c_4 ne peuvent pas apprendre leur point de sortie optimal c_2.

La figure 1 illustre un cas de routage sous-optimal. ρ et ρ′ sont deux routes concurrentes quasi-équivalentes respectivement apprises par les routeurs c_1 et c_2. Lorsque rr compare ces deux routes, il privilégie ρ′ puisque $|rr,c_1| < |rr,c_2|$. À terme, c_3 et c_4 n'ont donc connaissance que de ρ′, et n'apprennent jamais l'existence de leur point de sortie optimal (c_2). En conséquence, ils expulsent le trafic à destination de D par c_1 via rr, illustrant la sous-optimalité du routage.

[BMU07, DW12] caractérisent l'ensemble des routeurs susceptibles de bloquer la propagation d'une route BGP étant donné un point de sortie $s \in \mathcal{S}$ vers un routeur cible $t \in \mathcal{T}$.

1. On extrait les points de sortie moins intéressants pour t que s : $\mathcal{C}(s,t) = \{s' \in \mathcal{S}, |t,s'| > |t,s|\}$
2. On colore en noir les routeurs qui préfèrent à s un point de sortie de $\mathcal{C}(s,t)$. Les autres routeurs sont colorés en blanc, et appartiennent à : $\mathcal{W}(s,t) = \{w \in \mathcal{T} | \forall s' \in \mathcal{C}(s,t), |w,s| < |w,s'|\}$

La figure 1 illustre le cas $(s,t) = (c_2,c_4)$. L'existence d'un chemin de propagation iBGP impliquant uniquement des routeurs $\mathcal{W}(s,t)$ est une condition nécessaire et suffisante pour garantir que t aura bien connaissance de son point de sortie optimal s.

‡. On omet ici les oscillations de routage liées au MED, qui peuvent se contourner avec les options `set-deterministic-med` ou `always-compare-med`.

3 iBGP2, une redistribution iBGP idéale

L'idée clé d'iBGP2 consiste à redistribuer les informations iBGP conformément aux plus courts chemins IGP. Plus précisément, un routeur u retransmet une route relative à un point de sortie s à un voisin v si et seulement si u appartient à un plus court chemin de v à s (*critère de diffusion iBGP2*). En pratique, iBGP2 consiste donc à activer ou désactiver des sessions iBGP2 de sorte à ce qu'elles coïncident avec les liens IGP, et d'autre part à adapter les filtres installés sur ces sessions de sorte à ce qu'elles se conforment au critère de diffusion iBGP2.

Pour que u puisse déterminer si une route est pertinente ou pas pour v, il doit pouvoir calculer les plus courts chemins de v vers le point de sortie de la route. C'est pourquoi iBGP2 requiert un IGP à états de lien. Ainsi, u connaît tout le graphe IGP et peut calculer l'algorithme de Dijkstra du point de vue de ses voisins (à chaque changement de topologie IGP). Il peut ensuite corriger en conséquence les filtres associés à chaque voisin de sorte à ne propager que des routes satisfaisant le critère de diffusion iBGP2.

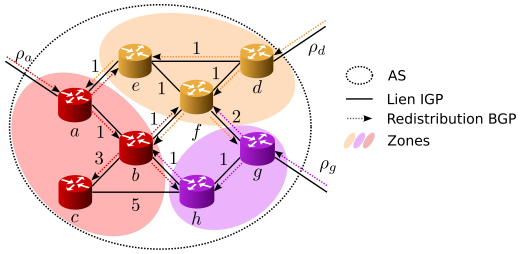

FIGURE 2: Exemple de redistribution iBGP2 avec trois routes concurrentes quasi-équivalentes.

La figure 2 illustre la manière dont iBGP2 diffuse les routes BGP une fois que le routage a convergé.

Optimalité et cohérence du routage : on peut montrer qu'iBGP2 conduit toujours à un routage fm-optimal. Soit s un point de sortie optimal pour v (on prendra par exemple $(s,t) = (a,c)$ et $(u,v) = (b,c)$).

— Par principe d'optimalité de Dijkstra, s est a fortiori un point de sortie optimal pour tout routeur u situé sur un plus court chemin de v à s. Ainsi $u \in \mathcal{W}(s,t)$.

— Il existe toujours un chemin de propagation iBGP2 entre s et v s'ils sont dans la même composante connexe IGP (a,b,c dans notre exemple). En effet, tout plus court chemin de t à s n'implique que des routeurs de $\mathcal{W}(s,t)$. Le chemin iBGP2 inverse qui se superpose (ici c,b,a) vérifie le critère de propagation iBGP2 de bout en bout, donc t a bien connaissance de la route correspondant à s.

Robustesse : L'overlay iBGP2 s'adapte à tout instant à la structure du graphe IGP, il est donc par construction résilient aux changements de topologie IGP. De plus, il se met à jour aussi vite que l'IGP sous-jacent, donc en pratique très rapidement. Voyons à présent le comportement d'iBGP2 face aux changements BGP. On appelle "zones" les ensembles de routeurs adoptant un même point de sortie pour une destination donnée (représentées en rouge, orange et violet sur la figure 2). Seuls certains routeurs ont connaissance de points de sortie alternatifs (e.g. a,b pour la zone rouge). Cette *diversité de routes* leur permet de basculer rapidement sur leur nouveau meilleur point en cas de besoin. Les autres routeurs (ici c) doivent attendre d'apprendre une route alternative, et risquent dans l'intervalle, de perdre le trafic correspondant.

Passage à l'échelle : Chaque routeur maintient autant de sessions iBGP2 qu'il a de voisins IGP (peu en pratique). On s'attend donc à ce qu'iBGP2 ait un coût en mémoire et en CPU raisonnable.

4 Implémentation et évaluation par simulation

Nous avons réalisé une implémentation réaliste d'iBGP2 sur Quagga à l'aide de ns3-dce [§].

Les simulations ont été réalisées sur la topologie décrite par la figure 2. On compare le nombre de messages BGP échangés, la consommation en RAM, et la diversité de route obtenue avec un full-mesh iBGP (FM), iBGP2, et une topologie de RR (où b et f jouent le rôles des RR et partagent une session, et où les

§. https://github.com/ibgp2/ibgp2

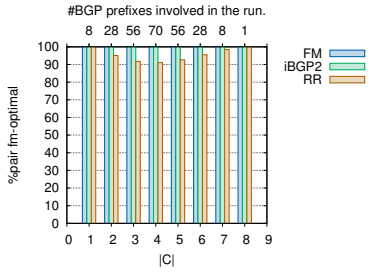

FIGURE 3: Fraction des paires (s,t) fm-optimales.

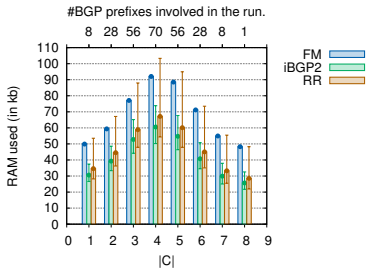

FIGURE 4: Consommation mémoire engendrée par `bgpd`.

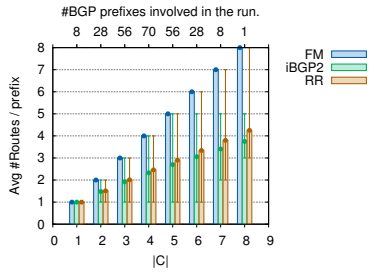

FIGURE 5: Nombre de point de sortie distincts appris (par routeur).

routeurs a,c,d,e,g,h jouent le rôle de leurs clients). Pour chaque ensemble $C \subseteq \mathcal{T}$, on construit un jeu de routes concurrentes quasi-équivalentes. On étudie le comportement de toutes les solutions dans toutes les situations possibles. Les résultats sont agrégés selon la taille de C. Les valeurs tracées correspondent aux valeurs moyennes, et les barres d'incertitudes à la valeur minimale et maximale observées pour $|C|$ donné.

Bien que l'instance considérée soit très petite, on retrouve les résultats attendus. La figure 3 montre que le full-mesh et iBGP2 amènent à un routage optimal, contrairement aux RR, et ce, malgré une topologie iBGP très maillée. Les figures 4 et 5 montrent qu'iBGP2 et les RR conduisent à une consommation mémoire et à une diversité moyenne de routes comparables. La figure 4 montre que parfois, un routeur peut consommer plus de mémoire dans l'approche RR qu'avec un full-mesh. En effet, les RR peuvent apprendre un point de sortie via plusieurs chemins iBGP. A contrario, cette redondance a rarement lieu avec iBGP2, car par construction, un routeur n'apprend un point de sortie que via un seul voisin la plupart du temps.

5 Conclusion

Dans cet article, nous avons présenté un nouveau mécanisme de redistribution de routes iBGP capable de passer à l'échelle, appelé iBGP2. Celui-ci permet de concilier les avantages du full-mesh (routage correct, réactif et optimal) et de la réflexion de route (passage à l'échelle), sans leurs inconvénients. Pour ce faire, iBGP2 se calque sur la topologie IGP. Il peut donc s'adapter à n'importe quelle topologie IGP et à n'importe quel changement de topologie IGP. Comme iBGP2 se reconfigure automatiquement, on élimine par la même occasion les risques d'erreur de configuration. Sur le plan opérationnel, iBGP2 peut être déployé dans un AS même si certains routeurs ne le supportent encore (il suffit de les configurer comme RR-clients de leurs voisins dans le graphe IGP). Enfin, iBGP2 peut se généraliser afin de gérer simultanément plusieurs plans de routage IGP. Il devient alors possible de choisir un point de sortie suivant différentes métriques, comme par exemple la bande passante ou le délai, en fonction de la nature de la destination.

Références

[BMU07] Marc-Olivier Buob, Mickael Meulle, and Steve Uhlig. Checking for optimal egress points in iBGP routing. In *Design and Reliable Communication Networks, 2007. 6th International Workshop on*, pages 1–8. IEEE, 2007.

[BUM08] Marc-Olivier Buob, Steve Uhlig, and Mickael Meulle. Designing optimal iBGP route-reflection topologies. In *NETWORKING 2008*, pages 542–553. Springer, 2008.

[DW12] Michael Dinitz and Gordon Wilfong. iBGP and constrained connectivity. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 122–133. Springer, 2012.

[GW02a] Timothy G. Griffin and Gordon Wilfong. Analysis of the MED oscillation problem in BGP. In *Network Protocols, 2002. Proceedings. 10th IEEE International Conference on*, pages 90–99. IEEE, 2002.

[GW02b] Timothy G. Griffin and Gordon Wilfong. On the correctness of iBGP configuration. *ACM SIGCOMM Computer Communication Review*, 32(4) :17–29, 2002.